REMARKS

With this amendment, claims 1, 3-16, 18-25, 27-33, and 37-40 are pending in the present application. Claims 1, 14, 16, and 30-33 have been amended to more particularly recite the claimed the invention. The amendments to claims 1, 14, 16, and 30-33 are supported in the specification at, for example, page 43, l. 11 through pg. 44, l. 2 and Figure 17. Claim 40 has been added and is believed to be allowable. Claim 40 is supported in the specification at, for example, page 43, l. 11 through pg. 44, l. 2 and Figure 17. Continued examination of the present application as amended is hereby requested.

The Applicant has carefully and thoughtfully considered the Office Action and the comments therein. For the reasons given below, it is submitted that this application is in condition for allowance.

Rejections Under 35 USC § 103(a) – Coad in view of Little

On pages 2-15, the Office Action rejects claims 1, 3-5, 7-16, 18-20, 22-25 and 27-39 as being unpatentable over U.S. Patent No. 6,851,105 to Coad et al. (Coad) in view of U.S. Patent No. 7,047,518 to Little et al. (Little). Applicants respectfully traverse the rejection. Coad and Little are discussed first, followed by the Action's rejections of claims 1, 3-5, 7-16, 18-20, 22-25 and 27-39.

A. Coad

Coad discloses "methods and systems for generating, applying, and defining patterns for existing code in order to improve the design and efficiency of the existing code." Coad, col. 1, l. 28-30 and col. 6, ll. 42-47. Patterns for software development ensure sound code architecture, help resolve common, recurring software development problems, and allow developers to organize, document, and produce more maintainable software. Coad, col. 6, l. 58-67.

Methods and systems disclosed by Coad utilize a software development tool that generates a pattern instance and applies the pattern instance to a portion of existing code to improve the design of the existing code. Coad, col. 6, ll. 42-47. The software development tool can also be used to create a new pattern from existing code. Coad, col. 6, ll. 42-47.

When improving the design of existing code, the software development tool performs three steps. First, the software development tool receives an element type to distinguish the type of pattern that is to be created. Coad, col. 7, l. 51-53 and Figure 2. Second, the software development tool displays a list of pattern options that are applicable to the selected element type and from which the developer can choose. Coad, col. 10, l. 1-3 and Figure 2. Third, once the developer has selected a pattern, the software generation tool applies the pattern to the code. Coad, col. 24, l. 64-67.

B. Little

Little discloses "a system for integrated computer software application development and modeling." Little, Abstract. The system provides tools to assist a software engineer in the refinement of a model by folding existing application and database knowledge into the model. Little, col. 9, l. 41-53. Particularly, "Design Patterns and Model Verification tools assist the software developer by performing much of the routine work necessary to complete a model." Little, col. 9, l. 53-56. For example, tools may apply a design pattern to the user's model in order to improve the model's integration and efficiency. Little, col. 10 37-40. Once the model is complete, "source code generation creates the implementation from the model and helps keep the model and source code synchronized." Little, col. 9, l. 56-58

C. Rejections of claims 1, 3-5, 7-16, 18-20, 22-25 and 27-39.

Initially, Applicants note that **the Action addresses two features that do not appear in claim 1** – namely "prompting a user to specify the embedded code" and "generating a code for a code generation code" – and are, therefore, not addressed in this response.

Regarding claim 1, Coad and Little, taken either singly or in any reasonable combination, do not disclose or render obvious the claimed invention for at least the following three reasons.

First, Coad does not disclose "at least one code generation goal being used to generate embedded code from the graphical model in a graphical modeling environment, the at least one code generation goal relating to a characteristic of the code to be generated from the graphical model," as recited in amended claim 1. Instead, Coad teaches applying a pattern to a portion of

existing code in order to improve the design and efficiency of the existing code." Coad, col. 6, ll. 42-47. Patterns ensure sound code architecture, help resolve common, recurring software development problems, and allow developers to organize, document, and produce more maintainable software. Coad, col. 6, l. 58-67. **Hence, Coad discloses applying a pattern to preexisting code, which is not equivalent to "at least one code generation goal being used to generate embedded code from the graphical model in a graphical modeling environment, the at least one code generation goal relating to a characteristic of the code to be generated from the graphical model," as recited in claim 1**. Therefore, Coad does not disclose "at least one code generation goal being used to generate embedded code from the graphical model in a graphical modeling environment, the at least one code generation goal relating to a characteristic of the code to be generated from the graphical model," as recited in claim 1.

Little does not overcome the failings of Coad. Instead, Little discloses "a system for integrated computer software application development and modeling." Little, Abstract. The system provides tools to assist a software engineer in the refinement of a model by folding existing application and database knowledge into the model. Little, col. 9, l. 41-53. Particularly, "Design Patterns and Model Verification tools assist the software developer by performing much of the routine work necessary to complete a model." Little, col. 9, l. 53-56. For example, tools may apply a design pattern to the user's model in order to improve the model's integration and efficiency. Little, col. 10 37-40. Once the model is complete, "source code generation creates the implementation from the model and helps keep the model and source code synchronized." Little, col. 9, l. 56-58. **Hence, Little discloses applying a design pattern to a user's model, which is not equivalent to "at least one code generation goal being used to generate embedded code from the graphical model in a graphical modeling environment, the at least one code generation goal relating to a characteristic of the code to be generated from the graphical model," as recited in claim 1.** Hence, the combination of Coad and Little fails to teach or render obvious "at least one code generation goal being used to generate embedded code from the graphical model in a graphical modeling environment, the at least one code generation goal relating to a characteristic of the code to be generated from the graphical model," as recited in claim 1.

Second, the Action did not address "prompting a user to specify at least one code generation goal from a plurality of code generation goals," as recited in claim 1. In particular, the Action relies on Coad to teach every feature of claim 1 except for generating code for a code generation goal, the at least one code generation goal being used to generate embedded code from the graphical model in a graphical modeling environment, and changing parameters of the graphical model that are inconsistent with the at least one code generation goal. Applicants agree. However, in addition to not teaching these three features of claim 1, Coad fails to disclose "prompting a user to specify at least one code generation goal from a plurality of code generation goals."

The Action discusses a limitation of "prompting a user to specify the embedded code" and states that it corresponds to "display[ing] pattern options corresponding to selected element type" in Figure 2 of Coad. However, as noted above, claim 1 does not recite "prompting a user to specify the embedded code." For the purposes of this response, the Action is assumed to have aligned the recited "prompting a user to specify at least one code generation goal from a plurality of code generation goals" of claim 1 with "display pattern options corresponding to selected element type" in Figure 2 of Coad.

Coad does not disclose "prompting a user to specify at least one code generation goal from a plurality of code generation goals," as recited in claim 1. Instead Coad discloses a "software development tool that generates [a] pattern instance and applies the pattern to a portion of existing code to improve the design of the existing code." Coad, col. 6, ll. 42-47. According to Coad, the software development tool may first receive an element type to distinguish the type of pattern that is to be created. Coad, col. 7, l. 51-53 and Figure 2. The software development tool may then display a list of pattern options that are applicable to the selected element type and from which the developer can choose. Coad, col. 10, l. 1-3 and Figure 2. The software generation tool may then apply the pattern to the code. Coad, col. 24, l. 64-67. That is, Coad discloses applying a pattern to preexisting code, which is not equivalent to the "code generation goal" of claim 1 that "relat[es] to a characteristic of the code to be generated from the graphical model." **Therefore, allowing a user to select a pattern to apply to existing code is not equivalent to "prompting a user to specify at least one code generation goal from a plurality of code generation goals," as recited in claim 1.** Therefore, Coad does not disclose

"prompting a user to specify at least one code generation goal from a plurality of code generation goals" from claim 1.

Little does not overcome the failings of Coad. Instead, Little discloses a system that provides tools to assist a software engineer in the refinement of a model by folding existing applications and database knowledge into the model. Little, col. 9, l. 41-53. The tools of Little may apply a design pattern to the user's model in order to improve the model's integration and efficiency. Little, col. 10 37-40. Once the model is complete, "source code generation creates the implementation from the model and helps keep the model and source code synchronized." Little, col. 9, l. 56-58. That is, Little discloses applying a design pattern to a user's model, which is not equivalent to the "code generation goal" of claim 1 which "relat[es] to a characteristic of the code to be generated from the graphical model." **Therefore, applying a design pattern to a user's model, as Little discloses, is not equivalent to "prompting a user to specify at least one code generation goal from a plurality of code generation goals," as recited in claim 1.** Hence, the combination of Coad and Little fail to teach or render obvious prompting a user to specify at least one code generation goal from a plurality of code generation goals," as recited in claim 1.

Third, the Action did not address "generating embedded code in accordance with the at least one code generation goal," as recited in claim 1. Instead, the Action aligns "generating code for a code generation goal," with the "application model" of Little. Little, col. 10, l. 25-55. As noted above, claim 1 does not recite "generating code for a code generation goal." However, for the purposes of this response, the Action is assumed to have aligned the recited "generating embedded code in accordance with the at least one code generation goal" with the "application model" of Little.

Little does not disclose "generating embedded code in accordance with the at least one code generation goal," as recited in claim 1. Instead, Little discloses a system that provides tools to assist a software engineer in the refinement of a model by folding existing applications and database knowledge into the model. Little, col. 9, l. 41-53. The tools of Little may apply a design pattern to the user's model in order to improve the model's integration and efficiency. Little, col. 10 37-40. Once the model is complete, "source code generation creates the

implementation from the model and helps keep the model and source code synchronized." Little, col. 9, l. 56-58 (emphasis added). However Little discloses applying a design pattern to a user's model, which is not equivalent to the "code generation goal" of claim 1 that "relat[es] to a characteristic of the code to be generated from the graphical model." **Therefore, generating code from a user's model, after a design pattern has been applied, as Little discloses, is not equivalent to "prompting a user to specify at least one code generation goal from a plurality of code generation goals," as recited in claim 1**. Further, Coad does not overcome the failings of Little. Hence, the combination of Coad and Little fail to disclose prompting a user to specify at least one code generation goal from a plurality of code generation goals," as recited in claim 1.

Therefore, Coad and Little, taken either singly or in any reasonable combination, fail to disclose or render obvious claim 1.

For reasons set forth above, Applicants believe that claim 1 is allowable and respectfully request that the above rejection of claim 1 be withdrawn and that claim 1 be allowed.

Dependent claims 2-13 depend on claim 1 and are believed to be allowable for at least the same reasons as above. Therefore, Applicants respectfully request that the above rejection of claims 2-13 be withdrawn and that claims 2-13 be allowed.

Independent claims 14, 16, and 30-33 recite subject matter similar to that recited in claim 1, which Applicants believe is allowable over Coad in view of Little. Therefore, Applicants believe claims 14, 16, and 30-33 are allowable for at least the reasons set forth above. Applicants respectfully request that the above rejection of claims 14, 16, and 30-33 be withdrawn and that claims 14, 16, and 30-33 be allowed.

Claim 14 recites "the code generation goal being used to generate embedded code from the graphical model, the code generation goal relating to a characteristic of the code to be generated from the graphical model," which is similar to "the at least one code generation goal being used to generate embedded code from the graphical model in a graphical modeling environment, the at least one code generation goal relating to a characteristic of the code to be generated from the graphical model" as recited in claim 1. Claim 14 recites "generating

embedded code in accordance with the code generation goal," which is similar to "generating embedded code in accordance with the at least one code generation goal," as recited in claim 1.

Claim 16 recites "the acquired at least one code generation goal being used to generate embedded code from the graphical model, the acquired at least one code generation goal relating to a characteristic of the code to be generated from the graphical model," which is similar to "the at least one code generation goal being used to generate embedded code from the graphical model in a graphical modeling environment, the at least one code generation goal relating to a characteristic of the code to be generated from the graphical model" as recited in claim 1. Claim 16 recites "generating embedded code in accordance with the acquired at least one code generation goal," which is similar to "generating embedded code in accordance with the at least one code generation goal," as recited in claim 1.

Claim 30 recites "the acquired at least one code generation goal being used to generate embedded code from the graphical model, the acquired at least one code generation goal relating to a characteristic of the code to be generated from the graphical model," which is similar to "the at least one code generation goal being used to generate embedded code from the graphical model in a graphical modeling environment, the at least one code generation goal relating to a characteristic of the code to be generated from the graphical model" as recited in claim 1. Claim 30 recites "one or more instructions for generating embedded code in accordance with the acquired at least one code generation goal," which is similar to "generating embedded code in accordance with the at least one code generation goal," as recited in claim 1.

Claim 31 recites "the code generation goal being used to generate embedded code from the graphical model, the code generation goal relating to a characteristic of the code to be generated from the graphical model," which is similar to "the at least one code generation goal being used to generate embedded code from the graphical model in a graphical modeling environment, the at least one code generation goal relating to a characteristic of the code to be generated from the graphical model" as recited in claim 1. Claim 31 recites "one or more instructions for generating embedded code in accordance with the code generation goal," which is similar to "generating embedded code in accordance with the at least one code generation goal," as recited in claim 1.

Claim 32 recites "the at least one code generation goal being used to generate embedded code from the graphical model in a graphical modeling environment, the at least one code generation goal relating to a characteristic of the code to be generated from the graphical model," which is similar to "the at least one code generation goal being used to generate embedded code from the graphical model in a graphical modeling environment, the at least one code generation goal relating to a characteristic of the code to be generated from the graphical model" as recited in claim 1. Claim 30 recites "one or more instructions for generating embedded code in accordance with the at least one code generation goal," which is similar to "generating embedded code in accordance with the at least one code generation goal," as recited in claim 1.

Claim 33 recites "a process for preparing a graphical model for a code generation process for creating code based on the graphical model and at least one code generation goal, wherein the at least one code generation goal relates to a characteristic of the code," which is similar to "the at least one code generation goal being used to generate embedded code from the graphical model in a graphical modeling environment, the at least one code generation goal relating to a characteristic of the code to be generated from the graphical model" as recited in claim 1. Claim 33 recites "wherein the computer program generates code in compliance with the at least one code generation goal," which is similar to "generating embedded code in accordance with the at least one code generation goal," as recited in claim 1.

Dependent claims 15, 18-25, 27-29 and 35-39 depend on claims 15, 18-25, 27-29 and 35-39, and therefore are believed to be allowable for at least the same reasons as above. Therefore, Applicants respectfully request that the above rejection of claims 15, 18-25, 27-29 and 35-39 be withdrawn and that claims 15, 18-25, 27-29 and 35-39 be allowed.

Added Claims

Claim 40 has been added and is supported in the specification at, for example, page 43, l. 11 through pg. 44, l. 2 and Figure 17. Claim 40 depends on claim 1 and is therefore allowable, at least, for being dependent from an allowable claim. Therefore, Applicants respectfully request that claim 40 be allowed.

CONCLUSION

All of the stated grounds of objection and rejection have been properly traversed, accommodated, or rendered moot. Applicants therefore respectfully request that the Examiner reconsider all presently outstanding objections and rejections and that they be withdrawn. Applicants believe that a full and complete reply has been made to the outstanding Office Action and, as such, the present application is in condition for allowance. If the Examiner believes, for any reason, that personal communication will expedite prosecution of this application, the Examiner is hereby invited to telephone the undersigned at the number provided.

Prompt and favorable consideration of this Amendment is respectfully requested.

Respectfully submitted,

Dated: 5/22/08

By _____

Michael A. Sartori, Ph.D.
Registration No.: 41,289
Kyle D. Petaja
Registration No.: 60,309
VENABLE LLP
P.O. Box 34385
Washington, DC 20043-9998
(202) 344-4000
(202) 344-8300 (Fax)
Attorney/Agent For Applicant

#935586